Claims

1. A method for generating an undeniable signature (y_1, ..., y_t) on a set of data, the method comprising the following steps:

- transforming the set of data (m) to a sequence of a predetermined number (t) of blocks (x_1, ... , x_t), these blocks being members of an Abelian group, this transformation being a one way function,

- applying to each block (x_i) a group homomorphism (f) to obtain a resulting value (y_i), in which a number of elements of an initial group (G) is larger than the number of elements (d) of a destination group (H).

2. The method of claim 1, wherein the initial group (G) is formed by invertible integers modulo n, i.e. Z_n^* .

3. The method according to claim 2, wherein the group homomorphism (f) computation is based on computation of a residue character (χ) on a set of elements Z_n^*.

4. The method according to claim 3, wherein the residue character (χ) computation in based on a parameter (π) serving as a key.

5. The method according to the claim 4, wherein this key parameter (π) is determined such as : $\pi \cdot \bar{\pi} = n$, $\bar{\pi}$ being the complex conjugate of π.

6. The method according to claim 2, wherein the group homomorphism (f) computation is determined in raising an element of Z_n^* to the power of r(q-1), in which $n = p \cdot q$ such that $p = rd + 1$ and q are prime, gcd(r, d) = 1, gcd(q - 1, d) = 1, then by computing a discrete logarithm.

7. The method according to claim 6, wherein the group homomorphism is calculated using a factorization of n.

8. The method according to claim 1, wherein the length of the signature is dependent of the number of elements of the destination group (d) and the number of blocks (t).

9. The method according to claim 4, wherein the parameter (π) is a secret key on an asymmetric key pair public/secret.

10. A Method of confirming by a Verifier an undeniable signature ($y_1, \ldots, y_t$) of a set of data (m) generated by a Signer taking into account a predefined security parameter (k) of the confirmation protocol, this Signer having a public/secret key pair, this method comprising the following steps:

- obtaining a personal value (ρ) from the Signer, this personal value being part of the public key ($G, H, d, \rho, (e_1, \ldots e_s)$) of the Signer,

- extracting a first sequence of elements ($e_1, \ldots e_s$) from the public key,

- generating a second sequence of elements ($g_1, \ldots g_s$) from the personal value (ρ),

- generating a third sequence of elements ($x_1, \ldots, x_t$) from the set of data (m),

- randomly picking challenge parameters $r_i \in G$ and $a_{ij} \in Z_d$ for $i = 1, \ldots, k$ and $j = 1, \ldots, s+t$ and computing a challenge value $u_i = dr_i + a_{i1}g_1 + \ldots a_{is}g_s + a_{is+1}y_1 + \ldots + a_{is+t}y_t$,

- sending by the Verifier the challenge value u_i to the Signer,

- receiving from the Signer a commitment value ($<v_i>$), this commitment value ($<v_i>$) being calculated by the Signer based on a response value $v_i = f(u_i)$,

- sending by the Verifier the challenge parameters r_i and a_{ij} to the Signer,

- verifying by the Signer whether $u_i = dr_i + a_{i1}g_1 + \ldots a_{is}g_s + a_{is+1}y_1 + \ldots + a_{is+t}y_t$, and in the positive event, opening by the Signer the commitment on the response value (v_i),

- verifying by the Verifier whether $v_i = a_{i1}e_1 + \ldots a_{is}e_s + a_{is+1}y_1 + \ldots + a_{is+t}y_t$.

11. A method for denying to a Verifier by a Signer on an alleged non-signature (z1, ..., zt) of a set of data (m) , this signature being supposedly generated according to claim 1 by the Signer, this Signer having a public/secret key pair, this method taking into account a predefined security parameter (ℓ) of the denial protocol and comprising the following steps:

- obtaining by the Verifier a personal value (ρ) of the Signer, this personal value being part of the public key ($G, H, d, \rho, (e_1, \ldots e_s)$) of the Signer,

- extracting by the Verifier a first sequence of elements ($e_1, \ldots e_s$) from the public key,

- generating by the Verifier and the Signer a second sequence of elements ($g_1, \ldots g_s$) from the personal value (ρ),

- generating by the Verifier and the Signer a third sequence of elements ($x_1, \ldots, x_i$) from the set of data (m),

- calculating by the Signer the true signature ($y_1, \ldots, y_t$),

- repeating the following steps ℓ times, ℓ being the predetermined security parameter,

  - randomly picking by the Verifier challenge parameters $r_j \in G$ and $a_{ji} \in Z_d$ for i = 1, ... , s and j = 1, ..., t and $\lambda \in Z_p$ where p is the smallest prime dividing d,

  - computing $u_j := dr_j + a_{j1}g_1 + \ldots a_{js}g_s + \lambda x_j$, and $w_j := a_{j1}e_1 + \ldots a_{js}e_s + \lambda z_j$ for j = 1 .. t,

  - sending by the Verifier the challenge values u_j and w_j to the Signer,

  - computing by the Signer a response test value $TV_j := (z_j-y_j)$.

  - for each j = 1 to t, determining whether the test value TV_j =0,

- in the negative event, calculating a test parameter λ_j according to the following formula : $w_j - v_{j,} = \lambda_j (z_j - y_j)$

- determining an intermediate value IV, this value being equal to one valid test parameter λ and in case of no valid test parameter is found, selecting as intermediate value a random value,

- sending a commitment value CT based on the intermediate value IV, to the Verifier,

- sending by the Verifier the challenge parameters r_j, a_{ji} and test parameter λ to the Signer,

- verifying by the Signer whether $u_j = dr_j + a_{j1}g_1 + \ldots a_{js}g_s + \lambda x_j$ and $w_j := a_{j1}e_1 + \ldots a_{js}e_s + \lambda z_j$ for $j = 1 .. t$ hold, in the positive event, the Signer opens the commitment on the intermediate value (IV) to the Verifier.

- verifying by the Verifier that the test parameter λ is equal to the intermediate value IV.

12. The method of claim 11, in which the determination of the valid test parameter comprises the check whether $(w_j - v_{j,})$ and $(z_j - y_j)$ are not equal to 0.

13. The method of claim 11, in which $j > 1$, the determination of the valid test parameter comprises the check whether $(w_j - v_{j,})$ and $(z_j - y_j)$ are not equal to 0, and that all of the test parameters are the same.